Sentiment Analysis FINAL

Joost Bloos

07/11/2021

```
#Install packages:
#Source: https://trinkerrstuff.wordpress.com/my-r-packages/qdap/

#if (!require("pacman")) install.packages("pacman")
#pacman::p_load(sentimentr, dplyr, magrittr)
#install.packages("devtools")
#install_github("trinker/qdapDictionaries")
#install_github("trinker/qdapRegex")
#install_github("trinker/qdapTools")
#install_github("trinker/qdap")
#install.packages("quanteda")
#install.packages("sentimentr")
#install.packages("ndjson")
#install.packages("NLP")
#install.packages("dplyr")
#install.packages("tidyr")
#install.packages("tm")
#install.packages("corpus")
#install.packages("syuzhet")
```

```
library(devtools)
```

```
## Loading required package: usethis
```

```
library(tm)
```

```
## Loading required package: NLP
```

```
library(qdap)
```

```
## Loading required package: qdapDictionaries
```

```
## Loading required package: qdapRegex
```

```
## Loading required package: qdapTools
```

```
## Loading required package: RColorBrewer
```

```
##
## Attaching package: 'qdap'

## The following objects are masked from 'package:tm':
##
##     as.DocumentTermMatrix, as.TermDocumentMatrix

## The following object is masked from 'package:NLP':
##
##     ngrams

## The following objects are masked from 'package:base':
##
##     Filter, proportions
```

```
library(sentimentr)
```

```
## Registered S3 methods overwritten by 'textclean':
##   method           from
##   print.check_text qdap
##   print.sub_holder qdap
```

```
library(ndjson)
```

```
##
## Attaching package: 'ndjson'

## The following object is masked from 'package:qdapRegex':
##
##     validate
```

```
library(corpus)
library(syuzhet)
```

```
##
## Attaching package: 'syuzhet'

## The following object is masked from 'package:sentimentr':
##
##     get_sentences
```

```
library(tidyr)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following object is masked from 'package:qdapTools':
##
##     id
```

```
## The following object is masked from 'package:qdapRegex':
##
##     explain

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(quanteda)
```

```
## Package version: 3.1.0
## Unicode version: 13.0
## ICU version: 69.1

## Parallel computing: 4 of 4 threads used.

## See https://quanteda.io for tutorials and examples.

##
## Attaching package: 'quanteda'

## The following object is masked from 'package:tm':
##
##     stopwords

## The following objects are masked from 'package:NLP':
##
##     meta, meta<-
```

```
library(ggplot2)
```

```
##
## Attaching package: 'ggplot2'

## The following object is masked from 'package:qdapRegex':
##
##     %+%

## The following object is masked from 'package:NLP':
##
##     annotate
```

```
#a good package, also takes into account  negative words and amplifiers
#see: http://www.inside-r.org/packages/cran/qdap/docs/polarity
```

```r
#getwd()
#setwd("C:/Ryerson University - Capstone project/Module 2/EIEEE - Large dataset/Combined")
```

```r
#Read in original data set May 2020
data_set_may <- read.csv("corona_tweets_59 May 2020", header = T, sep = ",")
```

```
## Warning in scan(file = file, what = what, sep = sep, quote = quote, dec = dec, :
## embedded nul(s) found in input
```

```r
#take a sample of 1,000, set seed to replicate results across several analysis of methods:
set.seed(1000)
rawData <- data_set_may[sample(nrow(data_set_may), size = 1000), ]
#write.csv(rawData,'rawData.csv')

str(rawData)
```

```
## 'data.frame':    1000 obs. of  35 variables:
##  $ coordinates               : chr  "" "" "" "" ...
##  $ created_at                : chr  "Sat May 16 23:31:16 +0000 2020" "Sat May 16 18:57:19 +0000 2020
##  $ hashtags                  : chr  "" "" "" "" ...
##  $ media                     : chr  "" "" "" "" ...
##  $ urls                      : chr  "" "" "" "https://www.nbcnews.com/now/video/officials-warn-chines
##  $ favorite_count            : int  0 0 0 0 0 0 0 0 1 1 ...
##  $ id                        : num  1.26e+18 1.26e+18 1.26e+18 1.26e+18 1.26e+18 ...
##  $ in_reply_to_screen_name   : chr  "" "" "" "" ...
##  $ in_reply_to_status_id     : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ in_reply_to_user_id       : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ lang                      : chr  "en" "en" "en" "en" ...
##  $ place                     : chr  "" "" "" "" ...
##  $ possibly_sensitive        : chr  "" "" "" "false" ...
##  $ quote_id                  : num  NA NA NA NA 1.26e+18 ...
##  $ retweet_count             : int  25 338 441 0 0 12022 4 11 1 0 ...
##  $ retweet_id                : num  1.26e+18 1.26e+18 1.26e+18 NA NA ...
##  $ retweet_screen_name       : chr  "business" "Suewilson91" "BreitbartNews" "" ...
##  $ source                    : chr  "<a href=\"http://twitter.com/download/iphone\" rel=\"nofollow\"
##  $ text                      : chr  "Many Americans have proven diligent in staying home to limit th
##  $ tweet_url                 : chr  "https://twitter.com/lemnosalt/status/1261801422430978048" "https
##  $ user_created_at           : chr  "Tue Feb 10 00:25:20 +0000 2009" "Sun Dec 01 15:12:16 +0000 2019
##  $ user_id                   : num  2.05e+07 1.20e+18 8.17e+17 1.66e+09 1.26e+18 ...
##  $ user_default_profile_image: chr  "false" "false" "false" "false" ...
##  $ user_description          : chr  "Groovy chick and media producer. All snark. No bite." "" "" "Jus
##  $ user_favourites_count     : int  92045 19675 1 46635 2788 1371 1230 18960 4 34505 ...
##  $ user_followers_count      : int  1469 45 65 263 426 97 109 2151 375 12607 ...
##  $ user_friends_count        : int  2526 229 228 1960 267 240 274 4846 227 12722 ...
##  $ user_listed_count         : int  73 0 1 1 4 1 0 15 13 106 ...
##  $ user_location             : chr  "" "New Forest" "" "United States" ...
##  $ user_name                 : chr  "Lynn" "Hilary ðŸ’\231" "Bill Spears" "Bet" ...
##  $ user_screen_name          : chr  "lemnosalt" "Hilary72926522" "BillSpears724" "Bet_the_ChE" ...
##  $ user_statuses_count       : int  35678 5272 24796 23697 1028 317 279 84594 14606 252203 ...
##  $ user_time_zone            : logi  NA NA NA NA NA NA ...
##  $ user_urls                 : chr  "http://lynnmargherita.com" "" "" "" ...
##  $ user_verified             : chr  "false" "false" "false" "false" ...
```

```
#create a corpus:
importdocs = corpus(rawData, text_field = 'text')
```

```
#preprocessing of data
importdocs <- gsub("'", "", importdocs)  # remove apostrophes
importdocs <- gsub("[[:punct:]]", " ", importdocs)  # replace punctuation with space
importdocs <- gsub("[[:cntrl:]]", " ", importdocs)  # replace control characters with space
importdocs <- gsub("^[[:space:]]+", "", importdocs) # remove whitespace at beginning of documents
importdocs <- gsub("[[:space:]]+$", "", importdocs) # remove whitespace at end of documents
importdocs <- tolower(importdocs)
```

```
mycorpus <- get_sentences(importdocs)
mysentiment <- sentiment(mycorpus)
mysentiment
```

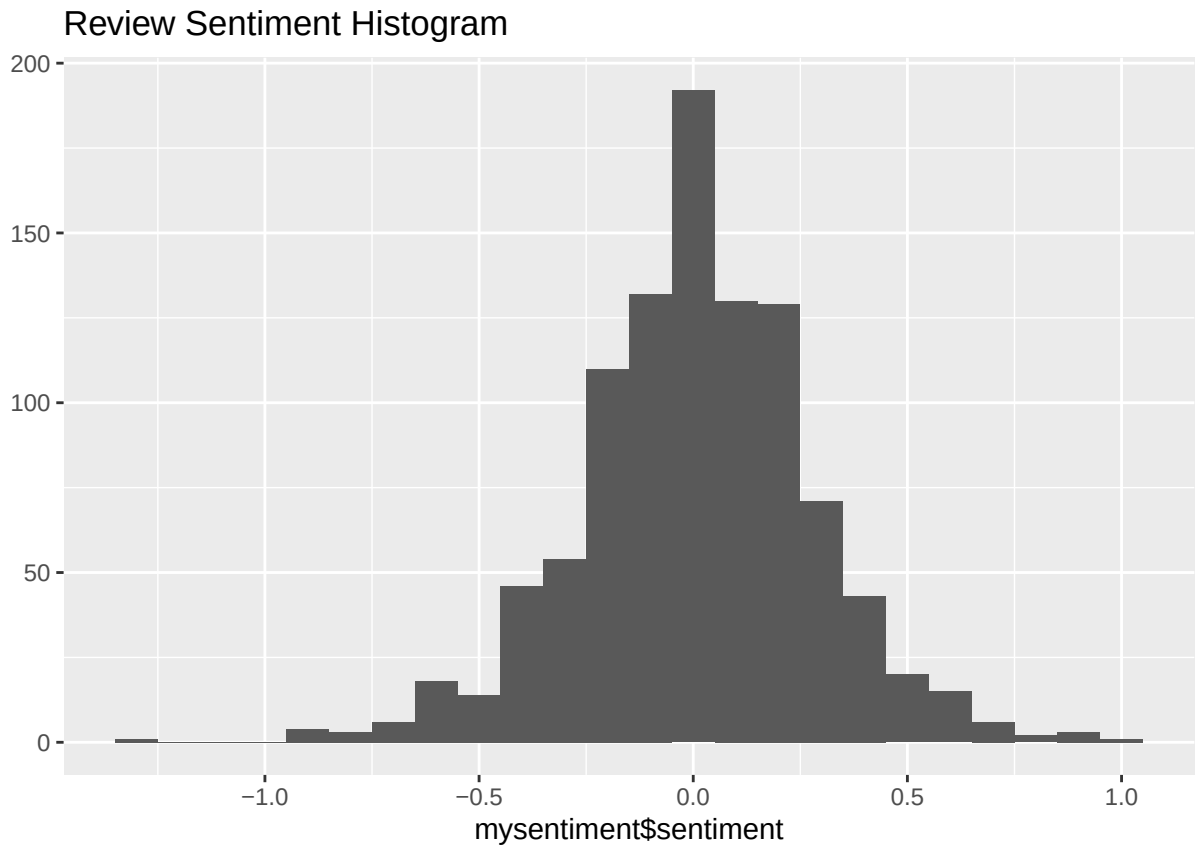
```
##       element_id sentence_id word_count  sentiment
##    1:          1           1         29  0.3992450
##    2:          2           1         36 -0.1063333
##    3:          3           1         45 -0.1192570
##    4:          4           1         17 -0.1819017
##    5:          5           1         25  0.1140000
##   ---
##  996:        996           1         45  0.1043498
##  997:        997           1         21  0.1963961
##  998:        998           1         13  0.2773501
##  999:        999           1         51 -0.2450490
## 1000:       1000           1         16  0.0000000
```

```
# run overall score, result overall neutral to perhaps moderate positive
summary(mysentiment$sentiment)
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## -1.319096 -0.154814  0.000000  0.005574  0.184188  0.952206
```

```
#results expressed in histogram

qplot(mysentiment$sentiment,   geom="histogram",binwidth=0.1,main="Review Sentiment Histogram")
```




```
#source: https://www.programmingr.com/sentiment-analysis/

#returns the individual words along with their polarity strength and counts.
t = extract_sentiment_terms(mycorpus)
attributes(t)$count
```

```
##             words polarity  n
##    1:        care     1.00 38
##    2:      please     1.00 22
##    3:       truth     1.00  9
##    4:  understand     1.00  8
##    5:         top     1.00  8
##   ---
## 7155:  could have    -1.05  3
## 7156: should have    -1.05  2
## 7157:    too many    -2.00  7
## 7158:      i wish    -2.00  2
## 7159:    too much    -2.00  2
```

```
#show positive and negative word use:
head(t,20)
```

```
##     element_id sentence_id                       negative
##  1:          1           1                   limit,waning
##  2:          2           1 accused,liar,resign,lies,liars
```

```
##  3:          3           1       trump,communist,pandemic
##  4:          4           1                           warn
##  5:          5           1                concern,pandemic
##  6:          6           1                     untrue,deny
##  7:          7           1
##  8:          8           1                       bad,havoc
##  9:          9           1
## 10:         10           1                         threaten
## 11:         11           1                              cut
## 12:         12           1
## 13:         13           1
## 14:         14           1  coroner,death,death,poisoning
## 15:         15           1
## 16:         16           1
## 17:         17           1
## 18:         18           1
## 19:         19           1
## 20:         20           1               ignorant,meltdown
##                         positive
##  1:   proven,diligent,acceptance
##  2:        protected,care,league
##  3:                  accountable
##  4:
##  5:                    like,good
##  6:            care,support,like
##  7: obtaining,results,technology
##  8:
##  9:
## 10:
## 11:                work,freedom
## 12:                        safe
## 13:                    positive
## 14:  determined,content,measured
## 15:
## 16:                     working
## 17:           confirmed,positive
## 18:                    good,work
## 19:                        care
## 20:              flatter,flatter
```

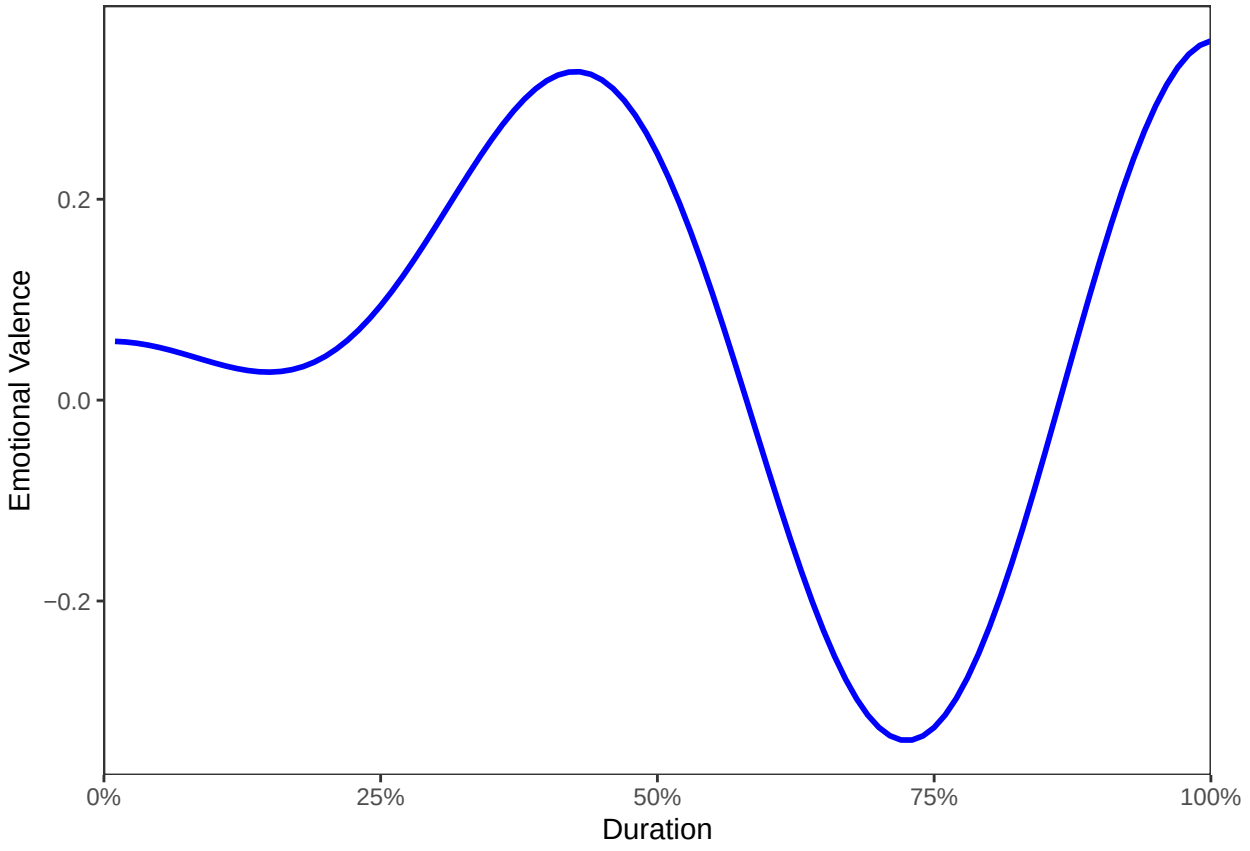
```
# The emotion() function returns the rate of emotion per sentence. A data frame is returned by this fun
emotion(mycorpus[1:2])
```

```
##     element_id sentence_id word_count         emotion_type emotion_count
##  1:          1           1         29                anger             0
##  2:          1           1         29         anticipation             0
##  3:          1           1         29              disgust             0
##  4:          1           1         29                 fear             0
##  5:          1           1         29              sadness             0
##  6:          1           1         29                trust             1
##  7:          1           1         29        anger_negated             0
##  8:          1           1         29 anticipation_negated             0
##  9:          1           1         29      disgust_negated             0
## 10:          1           1         29         fear_negated             0
```

```
## 11:          1           1         29                 joy             0
## 12:          1           1         29         joy_negated             0
## 13:          1           1         29      sadness_negated            0
## 14:          1           1         29             surprise            0
## 15:          1           1         29     surprise_negated            0
## 16:          1           1         29        trust_negated            0
## 17:          2           1         36                anger            2
## 18:          2           1         36         anticipation            1
## 19:          2           1         36              disgust            2
## 20:          2           1         36                 fear            2
## 21:          2           1         36              sadness            1
## 22:          2           1         36                trust            1
## 23:          2           1         36        anger_negated            0
## 24:          2           1         36 anticipation_negated            0
## 25:          2           1         36      disgust_negated            0
## 26:          2           1         36         fear_negated            0
## 27:          2           1         36                  joy            0
## 28:          2           1         36          joy_negated            0
## 29:          2           1         36      sadness_negated            0
## 30:          2           1         36             surprise            0
## 31:          2           1         36     surprise_negated            0
## 32:          2           1         36        trust_negated            0
##     element_id sentence_id word_count         emotion_type emotion_count
##        emotion
##  1: 0.00000000
##  2: 0.00000000
##  3: 0.00000000
##  4: 0.00000000
##  5: 0.00000000
##  6: 0.03448276
##  7: 0.00000000
##  8: 0.00000000
##  9: 0.00000000
## 10: 0.00000000
## 11: 0.00000000
## 12: 0.00000000
## 13: 0.00000000
## 14: 0.00000000
## 15: 0.00000000
## 16: 0.00000000
## 17: 0.05555556
## 18: 0.02777778
## 19: 0.05555556
## 20: 0.05555556
## 21: 0.02777778
## 22: 0.02777778
## 23: 0.00000000
## 24: 0.00000000
## 25: 0.00000000
## 26: 0.00000000
## 27: 0.00000000
## 28: 0.00000000
## 29: 0.00000000
## 30: 0.00000000
```

```
## 31: 0.00000000
## 32: 0.00000000
##        emotion
```

```
# graph with emotional valence, what is explanation. Note to self: look up
plot(mysentiment)
```




```
#integrate sentiment score into updated dataset
sentimentResultMay2020 <- rawData
sentimentResultMay2020$sentiment_score = mysentiment$sentiment
str(sentimentResultMay2020)
```

```
## 'data.frame':    1000 obs. of  36 variables:
##  $ coordinates               : chr  "" "" "" "" ...
##  $ created_at                : chr  "Sat May 16 23:31:16 +0000 2020" "Sat May 16 18:57:19 +0000 2020
##  $ hashtags                  : chr  "" "" "" "" ...
##  $ media                     : chr  "" "" "" "" ...
##  $ urls                      : chr  "" "" "" "https://www.nbcnews.com/now/video/officials-warn-chines
##  $ favorite_count            : int  0 0 0 0 0 0 0 0 1 1 ...
##  $ id                        : num  1.26e+18 1.26e+18 1.26e+18 1.26e+18 1.26e+18 ...
##  $ in_reply_to_screen_name   : chr  "" "" "" "" ...
##  $ in_reply_to_status_id     : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ in_reply_to_user_id       : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ lang                      : chr  "en" "en" "en" "en" ...
##  $ place                     : chr  "" "" "" "" ...
```

```
##  $ possibly_sensitive        : chr  "" "" "" "false" ...
##  $ quote_id                  : num  NA NA NA NA 1.26e+18 ...
##  $ retweet_count             : int  25 338 441 0 0 12022 4 11 1 0 ...
##  $ retweet_id                : num  1.26e+18 1.26e+18 1.26e+18 NA NA ...
##  $ retweet_screen_name       : chr  "business" "Suewilson91" "BreitbartNews" "" ...
##  $ source                    : chr  "<a href=\"http://twitter.com/download/iphone\" rel=\"nofollow\"
##  $ text                      : chr  "Many Americans have proven diligent in staying home to limit th
##  $ tweet_url                 : chr  "https://twitter.com/lemnosalt/status/1261801422430978048" "http
##  $ user_created_at           : chr  "Tue Feb 10 00:25:20 +0000 2009" "Sun Dec 01 15:12:16 +0000 2019
##  $ user_id                   : num  2.05e+07 1.20e+18 8.17e+17 1.66e+09 1.26e+18 ...
##  $ user_default_profile_image: chr  "false" "false" "false" "false" ...
##  $ user_description          : chr  "Groovy chick and media producer. All snark. No bite." "" "" "Ju
##  $ user_favourites_count     : int  92045 19675 1 46635 2788 1371 1230 18960 4 34505 ...
##  $ user_followers_count      : int  1469 45 65 263 426 97 109 2151 375 12607 ...
##  $ user_friends_count        : int  2526 229 228 1960 267 240 274 4846 227 12722 ...
##  $ user_listed_count         : int  73 0 1 1 4 1 0 15 13 106 ...
##  $ user_location             : chr  "" "New Forest" "" "United States" ...
##  $ user_name                 : chr  "Lynn" "Hilary ðŸ’\231" "Bill Spears" "Bet" ...
##  $ user_screen_name          : chr  "lemnosalt" "Hilary72926522" "BillSpears724" "Bet_the_ChE" ...
##  $ user_statuses_count       : int  35678 5272 24796 23697 1028 317 279 84594 14606 252203 ...
##  $ user_time_zone            : logi  NA NA NA NA NA NA ...
##  $ user_urls                 : chr  "http://lynnmargherita.com" "" "" "" ...
##  $ user_verified             : chr  "false" "false" "false" "false" ...
##  $ sentiment_score           : num  0.399 -0.106 -0.119 -0.182 0.114 ...
```

```
#identify text for max (positive) sentiment score
max(mysentiment$sentiment)
```

```
## [1] 0.952206
```

```
maxSentiment <- sentimentResultMay2020[which.max(sentimentResultMay2020$sentiment_score),]
maxSentiment$text
```

```
## [1] "To defeat #COVID19 and build a more sustainable and equitable world, we need communities to com
```

```
#identify text for min sentiment score
min(mysentiment$sentiment)
```

```
## [1] -1.319096
```

```
minSentiment <- sentimentResultMay2020[which.min(sentimentResultMay2020$sentiment_score),]
minSentiment$text
```

```
## [1] "Remember Trump's idiotic statement about too much testing showing too many infections?\n\nTrump
```

#write sentiment score to original dataset write.csv(sentimentResultMay2020,'sentimentResultMay2020.csv')